```
B: = 0;
C: = 0;
P: = 3;
if p < 3 then; {comando vazio (desnecessário)}
begin
    B: = 5;
    C: = 10
end;
B: = B + 1;
C: = C + 1;
```

Problema:

Passar para o PASCAL:

```
se A ≠ B
  então
    se C = D
      então A ← B;
      senão C ← B + 2;
            A ← B + 1;
    fim se;
    X: = 3;
fim se;
```

Solução:

```
if A <> B
   then begin
          if C = D
           then A: = B
           else begin
                  C: = B + 2;
                  A: = B + 1
                end;
          X: = 3
        end
```

e) REPETIÇÃO (*enquanto-faça*)

PORTUGOL

```
enquanto <condição> faça
   <lista de comandos>
fim enquanto;
```

PASCAL

```
while <expressão> do <comando>
```


Observe que, no PASCAL, seguindo a palavra *do*, podemos ter apenas um comando. Se houver necessidade de mais de um comando, será necessário utilizar um comando composto.

Por exemplo, para passar para o PASCAL:

```
enquanto I ≤ 18 faça
    A ← C + 1;
    B ← D;
    D ← A;
    I ← I + 1;
fim enquanto;
```

devemos fazer:

```
while I <= 18 do
  begin
     A: = C + 1;
     B: = D;
     D: = A;
     I: = I + 1;
  end
```

já para passar para o PASCAL:

```
enquanto J ≥ 10 faça
   J ← J – 1;
fim enquanto;
```

teremos:

```
while J >= 10 do J: = J – 1;
```

e) REPETIÇÃO (*repita-até*)

PORTUGOL

```
repita
   <lista de comandos>
até <condição>
```

PASCAL

```
repeat
   <seqüência de comandos>
until <expressão>
```

Neste caso, entre as palavras *repeat* e *until*, aparece uma <seqüência de comandos>, logo *não há necessidade* de se utilizar um comando composto.

Por exemplo, para passar para o PASCAL: